*mathematics*

MDPI

*Article*

# Meta-Heuristic Optimization and Keystroke Dynamics for Authentication of Smartphone Users

**El-Sayed M. El-Kenawy [1,*], Seyedali Mirjalili [2,3,*], Abdelaziz A. Abdelhamid [4,*], Abdelhameed Ibrahim [5], Nima Khodadadi [6] and Marwa M. Eid [7]**

[1] Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt
[2] Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, QLD 4006, Australia
[3] Yonsei Frontier Lab, Yonsei University, Seoul 03722, Korea
[4] Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt
[5] Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
[6] Department of Civil and Environmental Engineering, Florida International University, Miami, FL 33199, USA
[7] Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 11152, Egypt
* Correspondence: skenawy@ieee.org (E.-S.M.E.-K.); ali.mirjalili@torrens.edu.au (S.M.); abdelaziz@cis.asu.edu.eg (A.A.A.)

**Abstract:** Personal Identification Numbers (PIN) and unlock patterns are two of the most often used smartphone authentication mechanisms. Because PINs have just four or six characters, they are subject to shoulder-surfing attacks and are not as secure as other authentication techniques. Biometric authentication methods, such as fingerprint, face, or iris, are now being studied in a variety of ways. The security of such biometric authentication is based on PIN-based authentication as a backup when the maximum defined number of authentication failures is surpassed during the authentication process. Keystroke-dynamics-based authentication has been studied to circumvent this limitation, in which users were categorized by evaluating their typing patterns as they input their PIN. A broad variety of approaches have been proposed to improve the capacity of PIN entry systems to discriminate between normal and abnormal users based on a user's typing pattern. To improve the accuracy of user discrimination using keystroke dynamics, we propose a novel approach for improving the parameters of a Bidirectional Recurrent Neural Network (BRNN) used in classifying users' keystrokes. The proposed approach is based on a significant modification to the Dipper Throated Optimization (DTO) algorithm by employing three search leaders to improve the exploration process of the optimization algorithm. To assess the effectiveness of the proposed approach, two datasets containing keystroke dynamics were included in the conducted experiments. In addition, we propose a feature selection algorithm for selecting the proper features that enable better user classification. The proposed algorithms are compared to other optimization methods in the literature, and the results showed the superiority of the proposed algorithms. Moreover, a statistical analysis is performed to measure the stability and significance of the proposed methods, and the results confirmed the expected findings. The best classification accuracy achieved by the proposed optimized BRNN is 99.02% and 99.32% for the two datasets.

**Keywords:** meta-heuristic optimization; feature selection; keystroke dynamics; smartphone; authentication; Dipper Throated Optimization; Bidirectional Recurrent Neural Network

**MSC:** 68T20



**Citation:** El-Kenawy E.S.M.; Mirjalili, S.; Abdelhamid, A.A.; Ibrahim, A.; Khodadadi, N.; Eid, M.M. Meta-Heuristic Optimization and Keystroke Dynamics for Authentication of Smartphone Users. *Mathematics* **2022**, *10*, 2912. https://doi.org/10.3390/math10162912

Academic Editors: Petr Stodola and Ioannis G. Tsoulos

Received: 15 July 2022
Accepted: 10 August 2022
Published: 13 August 2022

## 1. Introduction

Devices of the Internet-of-Things (IoT), such as smartphones, are expected to play a significant role in the future of intelligent cities as crowd-sensing entities [1]. Increased smartphone usage necessitates greater security and privacy protection measures. Personal Identification Numbers (PIN) and unlock patterns are popular authentication techniques for cellphones because of their high level of security. However, PINs have just four or six characters. Therefore, they are subject to shoulder-surfing attacks and are not as secure as other authentication techniques. Several biometric recognition approaches have been developed to overcome this constraint and authenticate a typical users using their unique biometric information. It is possible for a smartphone to learn unique information about its user, such as the user's face [2], fingerprint [3], or iris [4] data; PIN-based authentication, on the other hand, is required as the final step in the authentication process if the user cannot correctly enter their biometric recognition information or reaches the maximum number of attempts previously defined. If an intruder obtains knowledge of a user's Personal Identification Number (PIN), the smartphone will grant the intruder access during the PIN-entry phase.

When a person types in their PIN, their typing patterns are analyzed and used to classify them according to the keystroke-dynamics-based authentication system. Researchers have been successful in preventing such breaches of security. In addition, several studies have been conducted to improve the classification performance of these methods. These studies have focused on diversifying the features extracted from typing patterns, employing artificial rhythms to facilitate user classification, or utilizing an appropriate classifier for binary classification. Over the past few decades, multiple optimization approaches have been gradually developed for handling challenging optimization problem in different fields. The optimization algorithms are based on various inspirations from different aspects such as nature [5], art [6], and physics [7], which are applied for different fields of study [8–10]. In addition, meta-heuristic algorithms are used for the parameter optimization of some methods.

In contrast to currently used methodologies, the method proposed in this paper was aimed to increase the classification performance of the regular user by utilizing the meta-heuristic optimization applied to the Bidirectional Recurrent Neural Network (BDRNN). The parameters of BDRNN are optimized using a new optimization algorithm based on the Dipper Throated Optimization (DTO) with dynamic weights assigned to three search leaders instead of one to improve the optimization results. Following feature selection and preprocessing, the data obtained from two keystroke dynamics datasets were employed for assessing the proposed algorithm. The achieved results were then compared to those of the other optimization and machine learning techniques to prove the superiority of the proposed approach. The findings showed that the proposed method improved user categorization in analytical and practical aspects.

The remainder of this paper is organized as follows. Section 2 presents the literature review of the previous studies on user authentication using keystroke dynamics. Then, Section 3 discusses the keystroke dynamics used to collect and extract feature data in order to increase classification accuracy. The proposed optimization methodology is discussed in detail in Section 4, which focuses on the proposed algorithm and its guiding concepts. The conducted experiments and the achieved results are detailed in Section 5, where they are also contrasted and examined in terms of performance. The conclusions and future research areas are presented in Section 6.

## 2. Literature Review

The previous studies on user authentication and categorization using keystroke dynamics are covered in this section.

Users are progressively storing sensitive information (such as passwords of bank accounts, text messages, and PIN codes) on smartphones as they grow more integrated into our everyday lives. Research on the PIN entry method using random keypads rather

than the current conventional keypads has been carried out to safeguard and protect such sensitive information from various dangers. A random keypad in a mobile setting was used to test the time it required to finish four and eight digit numbers in a research work from 2010 [11]. This system's accuracy was compared to that of a standard keypad by counting how many keystrokes were correctly made. When entering four-digit numbers, the random keypad took nearly a second longer to complete each input, yet the mistake rate was just 0.01% higher, indicating a better performance. There was a 0.03% increase in the mistake rate while typing eight-digit figures, which took an additional 3 seconds to complete each input.

A common form of behavioral biometrics is "keystroke-dynamics-based authentication," and it works by recognizing users based on the patterns and rhythms of their typing. This information is obtained while the users are typing on computer keyboards or the touchscreens of their smartphones. Since 1975, the practice of authenticating a person using keystroke dynamics has been in use [12], when this notion was initially introduced as an approach to identifying a user based on their typing habits. Classes with a False Rejection Rate (FRR) and a False Acceptance Rate (FAR) of 12% and 6%, respectively, were achievable in classification tests using this keystroke-based approach in 1985 [13]. Measured and retrieved from keyboard inputs, the amount of time between each keystroke was measured and used as a factor in the authentication process.

Keystroke-data-based user categorization performance can be improved by increasing the number of training datasets and normalizing the data gathered [14]. After having each of the thirty people who participated in the test enter 10 characters 20 times, the resulting data were normalized using user-dependent and user-independent methods. To complete each classification test successfully, it was essential to use varying amounts of practice data (5 to 10 data points out of the 20 collected data for each subject). The best results were obtained when the number of training data sets for each individual was set to 10, with an Equal Error Rate (EER) of 14.46%. When it comes to user authentication apps, this keystroke dynamics notion has been applied both on a PC and mobile device. Mobile phone numbers and four-digit PINs were classified using keystroke dynamics in a smartphone authentication investigation conducted in 2002 [15]. It was used as a data feature to determine the average amount of time required to input all of the specified keys and the standard deviation of the amount of time and the amount of delay time that occurred between each keystroke, achieving an EER of 15% for the 4-digit PINs.

In a study conducted in 2018, researchers attempted to identify people based on data collected by a smartphone's motion sensor [16]. It was found that the motion data had a significant impact on the outcomes. Three different types of motion sensors were included in this design, namely, acceleration sensors, angular velocity sensors, and rotation-vector sensors. Calculations were performed based on the data collected by each sensor in order to determine the averages, standard deviations, sums of positive and negative integers, and the root mean square. This approach achieved an EER of 8.94% using user categorization findings without motion sensor information. The mean feature, which is based on motion sensors, decreased that amount by 1.05% to 7.89%. The motion sensor-based features performed best with the mean feature. These findings showed that using motion sensor feature data to enhance user categorization performance was achievable.

It has recently been used in various contexts in several studies. Keystroke dynamics studies from free text, as opposed to those that use fixed-length elements such as passwords or PINs, are among them [17,18]. Others include studies that continuously classify a user's keystroke dynamics [18–20] and studies that take into account different types of typing postures such as relaxing positions, walking, and sitting [21]. There has been a lot of interest in biometric-based user authentication in addition to keystroke dynamics-based authentication. Recent attention has been paid to authentication methods based on electroencephalogram information as part of the attempt to address the limitations of existing biometric authentication systems, such as biometric information forgery [22] and Electrocardiogram (ECG) [23] information.

Keypad side-channel keystroke inference attacks may be mitigated using several techniques, such as [24]. The keypad buttons might be rearranged and increased in size, for example. The following are some effective defenses against side-channel keystroke inference attacks that were proposed in the study: Individual Key Randomization (IKR), Column Randomization (CR), Row Randomization (RR), and gray-scale IKR are all examples of keypad button randomization.

As part of an experiment conducted in 2019, test respondents were asked to type fixed PINs into a random keypad that was supposed to vary every round [25]. Based on the acquired data, the performance of user categorization was evaluated and examined. A random keypad was used to generate 10 rounds of PINs for the 30 participants. As a result of the inputted information, a random forest classifier was used to gather 32 different features (such as the amount of time it takes to press and release a single button), which were then used to create a classification model. The EER was found to be 10% as a result. However, the previous research [11,25], in contrast to our study, which provides a keypad that gathers unique qualities that reflect just regular humans, random keypad approaches could not easily acquire data on special features to categorize normal users efficiently. As an extra precaution against malicious keystroke inference attacks, one research [24] looked at the effectiveness of randomized keypads. Their goal is to determine whether or not the usage of distinct keypads helps to distinguish normal users from aberrant ones in a more accurate fashion.

Keystroke data may be used to identify typical and problematic smartphone users, according to the current research. An experiment in which participants were requested to type the four-letter key "abcd" in order to collect time vector data in the form of duration and interval time took place in [26]. Finally, feature selection strategies were explored based on the results of a genetic algorithm. Particle Swarm Optimization (PSO) was used to gather time vector data from 24 participants in a 2007 research as they input the four-digit key "abcd" [27]. Methods for selecting the best features for categorization were investigated. Twenty-two test individuals had passwords that were distinct from one another in [28]. Mean and standard deviations were calculated for each of the features derived from the gathered data: digraph, latency, and duration. They sought to employ Ant Colony Optimization (ACO) to enhance user categorization performance based on the measured data. To test and compare classification performance based on data obtained from 27 participants in [29], a variety of feature selection approaches were used, including PSO, ACO, Genetic Algorithm (GA), and Gravitational Search Algorithm (GSA). It was found that the accuracy ranged from 88.9% to 92.8%, with an EER of 0.063%, to 0.078%, and to 0.059%. In [30], Several methods were tried to categorize users, including the most often typed n-graph selection, the fastest typed n-graph selection, the time stability typed n-graph sample, and the time variation typed n-graph model. The most typically written n-graph selection strategy produced the best user classification results. In a 2020 study, feature scores were computed and evaluated using the data's trimmed mean and variance coefficients. Low-scoring elements were eliminated to improve classification accuracy [31–33]. A summary of the relevant studies in the literature addressing the task of smartphone user authentication is presented in Table 1.

**Table 1.** Summary of the relevant articles published in the literature.

| Ref. | Methodology | Result |
|---|---|---|
| [12] | Keystroke dynamics | False FRR = 12%, FAR = 6% |
| [15] | Keystroke dynamics | EER = 15% |
| [16] | Motion sensors | EER = 8.94% |
| [25] | Random keypad | EER = 10% |
| [34] | Unique Keypad | EER = 4.15% |

## 3. Keystroke Dynamics

"Keystroke dynamics" refers to the typing patterns or action types that emerge when a user hits the keys on a computer or smartphone. Keystroke dynamics-based authentication is a method of classifying people based on their typing patterns. During the authentication phase, data can be acquired from the user's repetitive presses and releases of the smartphone screen. The time it takes to press and release each key, as well as the movement of the smartphone while the keys are pressed, are all examples of valuable data. Using the keystroke data acquired, it is possible to establish keystroke-dynamics-based authentication. Touch data and motion data may be used to separate keystroke data, which can then be analyzed based on various parameters.

### *3.1. Touch Information*

Touch data are frequently collected when users touch the smartphone screen (keyDown) and remove their fingers from the screen (keyUp). We need to know how long it takes between touches on the smartphone screen, how much pressure the finger applies, and where the finger is positioned on the screen when the smartphone key is pressed and released to collect this data.

#### 3.1.1. Dwell Time

The *Dwell Time* (*DT*) is a measure of how long it takes for a user to enter a single key, as shown in Figure 1. According to Equation (1), the *DT* may be deduced from the recorded keyDown and keyUp data:

$$N.Release - N.Press = DT_n \quad (1)$$

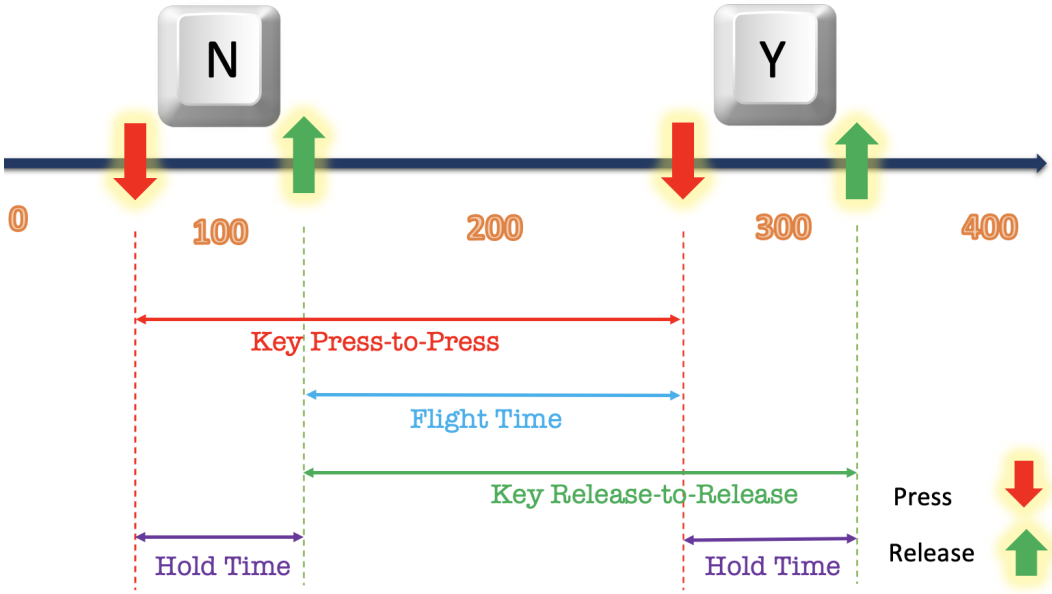


**Figure 1.** Time feature structure [34].

#### 3.1.2. Flight Time

When two keys are pressed, the *Flight Time* (*FT*) is calculated using the keyDown and keyUp data. Equation (2) is used to calculate the *FT* for each feature, whereas Equation (1) presents four *FT* features.

$$Flight\ Time = N.Release - Y.Press \quad (2)$$

#### 3.1.3. Pressure

Pressure data are captured when a user presses down on the screen and then lifts their finger off of it. A user's down pressure data correspond to their finger contacting the screen, while their up pressure data correspond to removing their finger from the screen.

#### 3.1.4. Coordinates

A coordinate data recorder is activated every time a user pushes on an area of the screen and then takes their finger off it. It is upX and upY features that are gathered when a user removes their finger from the screen; downX and downY are collected when a user pushes the screen.

#### 3.1.5. Motion Data

The smartphone is tracked while the user presses the keys on the screen to gather motion data. The x-, y-, and z-axis coordinates are used to measure the motion data. On the x-axis, the smartphone moves to the left and right, while on the y- and z-axes, the smartphone moves to the left or right, as illustrated in Figure 2. An acceleration, angular velocity and rotation vector are the three main categories of motion data that may be found in the data.

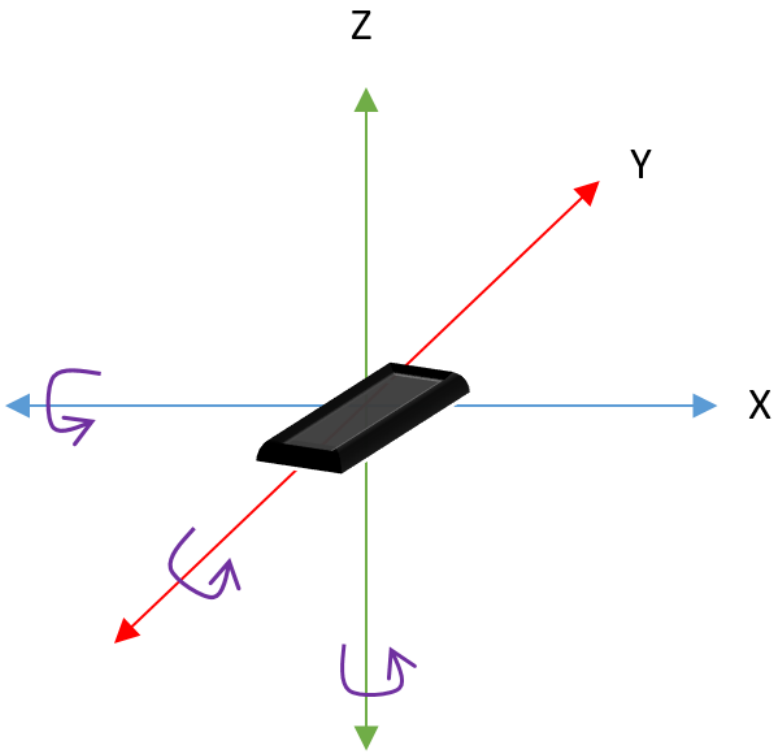


**Figure 2.** Motion data of smartphone reference axis [34].

#### 3.1.6. Accelerometer

Gravitational acceleration is most commonly detected via acceleration sensors, which may be used to estimate an object's tilt or vibration level. With this property, the acceleration is quantified. While not in use, a smartphone is still subjected to gravity's effects, which must be considered while extracting linear acceleration. The Accelerometer (acc) is used to denote acceleration, whereas Linear Acceleration (lacc) is denoted by the Linear Accelerometer (la).

#### 3.1.7. Angular Velocity

Angles per hour around a certain axis can be measured by using the angular velocity sensor. The angular velocity may be determined using this feature. Gyroscope is the unit of measure used to describe the rate of change in the angular velocity.

#### 3.1.8. Rotation Vector

The smartphone's rotational axes are depicted graphically by the rotation vectors. A geomagnetic sensor is used to determine the orientation of the smartphone. The north pole influences the rotation vectors detected by a geomagnetic sensor. Rotation vectors are computed instead of utilizing geomagnetic sensor measurements to remove the effect of the north pole. Just like in determining acceleration and angular velocity, the accelerometer and gyroscope data are employed. The letter rot denotes the rotation vectors, but the word game rotation represents the vectors for game rotation.

## 4. The Proposed Methodology

The proposed methodology is based on optimizing the parameters of Bidirectional Recurrent Neural Network (BRNN) using a modified Dipper Throated Optimization (DTO) algorithm. This section starts with presenting the basics of BRNN and DTO followed by presenting the proposed algorithm.

### *4.1. Bidirectional Recurrent Neural Network (BRNN)*

There are two parts to the concept of a Bidirectional Recurrent Neural Network. One part is responsible for forward states (forward neurons), and the other for backward states (backward states). Forward states' outputs are not coupled to backward states' inputs; the reverse is also true. According to Figure 3, this leads to the broad structure depicted in three time steps. This structure may be reduced to a typical unidirectional forward RNN without the backward states. They are reversing the time axis results in a regular RNN when forward states are removed. In contrast to the regular unidirectional RNN discussed above, the objective function can be directly minimized using input data from the past and future of the currently evaluated time frame without the need for delays to include future information because the same network handles both time directions citeBDRNN1, BDRNN2, BDRNN3. Figure 3 displays the typical structure of a BDRNN at three time steps, $t-1$, $t$, and $t+1$, for clarity. In the bottom section, information flows from left to right to represent the past, while in the top part, information flows from right to left to represent the future. $O_t$ is based on both the future and the past output symbolized by $h_t^b$ and $h_t^f$, respectively, at the process of the computation. In order to train BRNN, a specified time period's input information is used.

BDRNN can be trained using the same techniques as the traditional RNN because the two types of neurons in the BRNN architecture do not interact. BRNN's feed-forward structure is unfolded as a result of this process. A more complicated process is required when using back-propagation since the state and output changes cannot be performed one at a time. A Multi-Layer Perceptron (MLP) network may be updated in a similar way to the Back-Propagation Through Time (BPTT) networks. For the forward $t = 1$ and backward $t = T$ states, the values of the inputs are set to 0.5. The values of the forward and backward local state derivatives at $t = T$ and $t = 1$ states, on the other hand, are set to zero since it is assumed that information beyond this point is not important for the update carried out at the present state.

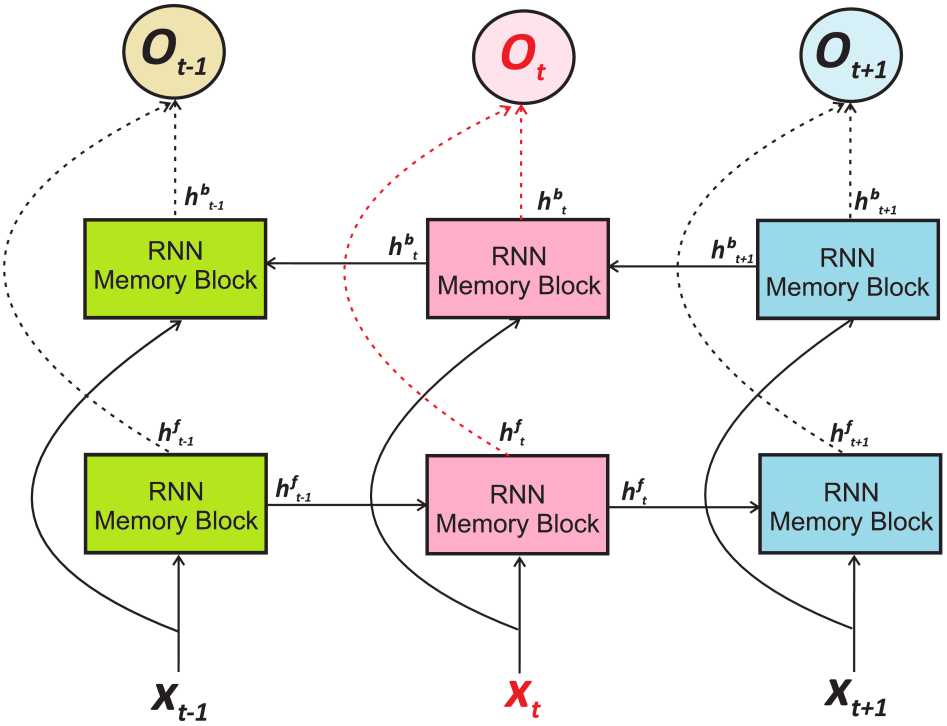


**Figure 3.** Typical Structure of Bidirectional Recurrent Neural Network shown at step three.

### *4.2. Dipper Throated Optimization (DTO)*

It is assumed that a flock of birds is swimming in search of food through the DTO algorithm. Each of the following matrices can be used to represent the birds' locations (represented by a matrix denoted by $P$) and velocities (represented by a matrix denoted by $V$) [35–37]. DTO can search for the best solution in the given search space based on these measures. The detailed steps of the traditional DTO algorithm are presented in Algorithm 1. The following matrices give more insight into the calculations of the DTO algorithm.

**Algorithm 1** The Dipper Throated Optimization algorithm

**Input** population, size, fitness function
**Output** best agent position
**Initialize** positions $P_i(i = 1, 2, \ldots, n)$, velocities $V_i(i = 1, 2, \ldots, n)$, and fitness function $h$.
**Initialize** iterations $M_t, r_1, r_2, K_1, K_2, K_3, K_4, K_5, R, t = 1$
**Get** $h$ for each agent $P_i$
**Find** best solution $P_{best}$
**while** $t \leq M_t$ **do**
  **for** $(i = 1 : i < n + 1)$ **do**
    **if** $R < 0.5$ **then**
      **Update** agent position
    **else**
      **Update** agent velocity
      **Update** agent position
    **end if**
  **end for**
  **Get** $h$ for each agent $P_i$
  **Update** $K_1, K_2, R$
  **Set** $P_{Gbest} = P_{best}$
**end while**
**Return** $P_{Gbest}$

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & \dots & P_{1,d} \\ P_{2,1} & P_{2,2} & P_{2,3} & \dots & P_{2,d} \\ P_{3,1} & P_{3,2} & P_{3,3} & \dots & P_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ P_{m,1} & P_{m,2} & P_{m,3} & \dots & P_{m,d} \end{bmatrix} \tag{3}$$

$$V = \begin{bmatrix} V_{1,1} & V_{1,2} & V_{1,3} & \dots & V_{1,d} \\ V_{2,1} & V_{2,2} & V_{2,3} & \dots & V_{2,d} \\ V_{3,1} & V_{3,2} & V_{3,3} & \dots & V_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ V_{m,1} & V_{m,2} & V_{m,3} & \dots & V_{m,d} \end{bmatrix} \tag{4}$$

where the $i$th bird in the $j$th dimension is denoted by $P_{(i,j)}$ for $i \in 1, 2, 3, \dots, m$ and $j \in 1, 2, 3, \dots, d$. The bird's speed in the $j$th dimension for $i \in 1, 2, 3, \dots, m$ and $j \in 1, 2, 3, \dots, d$ is referred to as $V_{(i,j)}$. There is a uniform distribution of the initial positions of $P_{(i,j)}$. For each bird, the values of the fitness functions $f = f_1, f_2, f_3, \dots, f_n$ are determined using the array below:

$$f = \begin{bmatrix} f_1(P_{1,1}, P_{1,2}, P_{1,3}, \dots, P_{1,d}) \\ f_2(P_{2,1}, P_{2,2}, P_{2,3}, \dots, P_{2,d}) \\ f_3(P_{3,1}, P_{3,2}, P_{3,3}, \dots, P_{3,d}) \\ \dots \\ f_m(P_{m,1}, P_{m,2}, P_{m,3}, \dots, P_{m,d}) \end{bmatrix} \tag{5}$$

where each bird's quest for food is reflected in its fitness score, and the mother bird is the superior value. Sorting is performed by ascending the values. $P_{best}$ has been proclaimed the first-best solution. Normal birds $P_{nd}$ are meant to be used as follower birds. $P_{Gbest}$ has been named the world's best solution. The optimizer's first DTO technique for updating the swimming bird's position is based on the following equations that update the position and speed of the individuals in the population:

$$X = P_{best}(i) - K_1.|K_2.P_{best}(i) - P(i)| \tag{6}$$

$$Y = P(i) + V(i+1) \tag{7}$$

$$P(i+1) = \begin{cases} X & \text{if } R < 0.5 \\ Y & otherwise \end{cases}, \tag{8}$$

$$\begin{aligned} V(i+1) = K_3 V(i) + K_4 r_1 (P_{best}(i) - P(i)) \\ + K_5 r_2 (P_{Gbest} - P(i)) \end{aligned} \tag{9}$$

where $i$ is the iteration number in which $P(i)$ is the average bird position, $P_{best}(i)$ is the position of the best bird, and $V(i+1)$ is the bird's speed at iteration $i+1$. $K_1$, $K_2$, and $K_3$ are weight values, and $K_4$ and $K_5$ are constants. $r_1$, $r_2$, and $R$ are random values in the range $[0, 1]$.

### *4.3. The Proposed Dynamic Weighted DTO Algorithm*

The proposed modified DTO optimization algorithm is based on Dynamic Weighted DTO and is referred to as (DWDTO) algorithm. The proposed algorithm divides the population into two groups: the exploration group and the exploitation group. In the traditional DTO, the exploration group is based on only one leader solution that explores the search space for finding the best solution. In the proposed algorithm, the leader solution works in collaboration with three other solutions to improve the exploration group's performance and reach the best solution faster. The detailed steps of the proposed algorithm are listed in Algorithm 2, and the coming sections discuss the main parts of the proposed algorithm.

**Algorithm 2** The Proposed DWDTO Algorithm

**Input** population $X_i$, $(i = 1, 2, \ldots, n)$, size $n$, and objective function $F_n$
**Output** best agent position
**Initialize** DWDTO configuration parameters
**Function** Fitness_Func (Solution $P$)
  **Calculate** and return fitness of $P$
**End** Function
**Set** $t = 1$
**while** $t <$ iter_Num **do**
  **Calculate** objective function $F_n$ for each agent $X_i$
  **Set** $N$ = best agent position
  **Randomize** $g_1, g_2, g_3$
  **Update** $\overrightarrow{f}(t+1)$ using Equation (10).
  **Decrease** $h$ exponentially from 1 to 0
  **In each group**: Update the number of solutions
  **if** the best fitness from the previous 4 iterations did not improve **then**
    **Increase** in the exploration group solutions number
  **end if**
  **for** each solution in the exploration group **do**
    **update** $fg_1, fg_2, fg_3$ and $N$
    The best solutions were elitism
    **if** $N <$ any of the best solutions **then**
      **Mutate** the solution by
      f(t+1) = $k + \frac{\sum((fg1)+(z*fgo2)+(k*fg3))}{2k}$
      $k = 1 - \frac{zt^2}{iters\_count^2}$
      k decreases exponentially from 2 to 0 over the course of iterations,
    **else**
      **Search** around current solution
      $P(i+1) = \begin{cases} P_{best}(i) - K_1.|K_2.P_{best}(i) - P(i)| & \text{if } R < 0.5 \\ P(i) + V(i+1) & otherwise \end{cases}$
    **end if**
  **end for**
  **for** each solution in the exploitation group **do**
    The best solutions were elitism
    **update** $fg_1, fg_2, fg_3$ and $N$
    **if** N < any of the best solutions **then**
      Move towards the best solution
      $f(t+1) = (g_1 * fo_1(t) + z * g_2 * (fg_2(t) - fg_3(t))) + (1-h) * g_3 * (leader(t) - xr1(t))$
    **else**
      **Search** around the best solution
      $V(i+1) = K_3 V(i) + K_4 r_1 (Pbest(i) - P(i)) + K_5 r_2 (PGbest - P(i))$
    **end if**
  **end for**
  **Update** fitness
  **Update** solutions
**end while**
**Return** best agent position $N$

#### 4.3.1. Exploration Group

This group is responsible for the exploration task for finding a promising point in the search space. In addition, it is responsible for avoiding becoming stuck in local optima, and to achieve that, the DWDTO uses two strategies. This step is applied to guarantee the population's diversity, allowing the DWDTO to search in different search spaces. In this

step, three random solutions are generated and guided by the leading solution to provide a better exploration of the search space. In this case, the objective function is measured using the following equations:

$$\begin{aligned}\overrightarrow{f}(t+1) = (\overrightarrow{g_1} * \overrightarrow{f}_{o1}(t) \\ + \overrightarrow{z} * \overrightarrow{g_2} \\ * (\overrightarrow{f}_{o2}(t) - \overrightarrow{f}_{o3}(t))) \\ + (1 - \overrightarrow{h}) * \overrightarrow{g_3} \\ * (leader(t) - \overrightarrow{f}_{o1}(t))\end{aligned} \tag{10}$$

where $h$ decreases exponentially from 1 to 0, and $fg_1$, $fg_2$, and $fg_3$ are three random solutions:

$$\overrightarrow{g_1} = rand(-1,1), \overrightarrow{g_2} = rand(0,1), \overrightarrow{g_3} = rand(-2,2). \tag{11}$$

The candidate searching around the promising areas in the search space is performed by finding the best fitness using the following equations:

$$\overrightarrow{D} = r_1(\overrightarrow{V}(t) - 1) \tag{12}$$

$$\overrightarrow{V}(t+1) = \overrightarrow{V}(t) + \overrightarrow{D}(2\overrightarrow{r}_2 - 1) \tag{13}$$

$$P(t+1) = \begin{cases} P_{best}(i) - K_1|K_2 P_{best}(i) - P(i)| & \text{if } R < 0.5 \\ P(i) + V(i+1) & otherwise \end{cases} \tag{14}$$

4.3.2. Exploitation Group

This group is responsible for finding the candidate solutions around the best solutions found so far. Searching around the best solution targets finding a much better solution. This search process is performed using the following formulation:

$$\overrightarrow{D} = \overrightarrow{P}(t) * (\overrightarrow{K} - r_4) \tag{15}$$

$$\overrightarrow{V}(t+1) = \overrightarrow{V}(t) + \overrightarrow{D}.(2\overrightarrow{r}_5 - 1) \tag{16}$$

$$\overrightarrow{K} = 1 - \frac{2k * Xt^2}{Solutions - count^2} \tag{17}$$

$$\begin{aligned}V(i+1) = K_3 V(i) + K_4 r_1 (P_{best}(i) \\ - P(i)) + K_5 r_2 (P_{Gbest} - P(i))\end{aligned} \tag{18}$$

where $i$ is the iteration number, in which $P(i)$ is the average bird position, $P_{best}(i)$ is the position of the best bird, and $V(i+1)$ is the bird's velocity at iteration $i+1$. $K_1$, $K_2$, and $K_3$ are weight values, and $K_4$ and $K_5$ are constants. $r_1$ and $r_2$ are random values in the range $[0,1]$.

4.3.3. Balance between Exploration and Exploitation

Maintaining the proper balance between exploration and extraction is critical to the success of the proposed DWDTO algorithm. After assigning 50% of the population to exploration and the other 50% to exploitation, the algorithm adjusts accordingly in the beginning. Figures 4 and 5 depict the dynamic balancing between the exploration and exploitation groups in the proposed approach. To begin with, having a large number of persons in the exploration group helps with discovering new areas of the search space. Exploitation individuals grow in quantity over time, whereas exploration persons decrease dynamically over time. This allows more people to enhance their overall fitness by allowing

additional exploitation of individuals to improve their overall fitness. It also utilizes an elitism method to keep the process's leader in place if no better solution can be found for fresh populations, assuring convergence. A DWDTO exploration group may expand to include more members when the leader's fitness level does not improve enough to prevent local optima and stagnation issues after three consecutive runs.

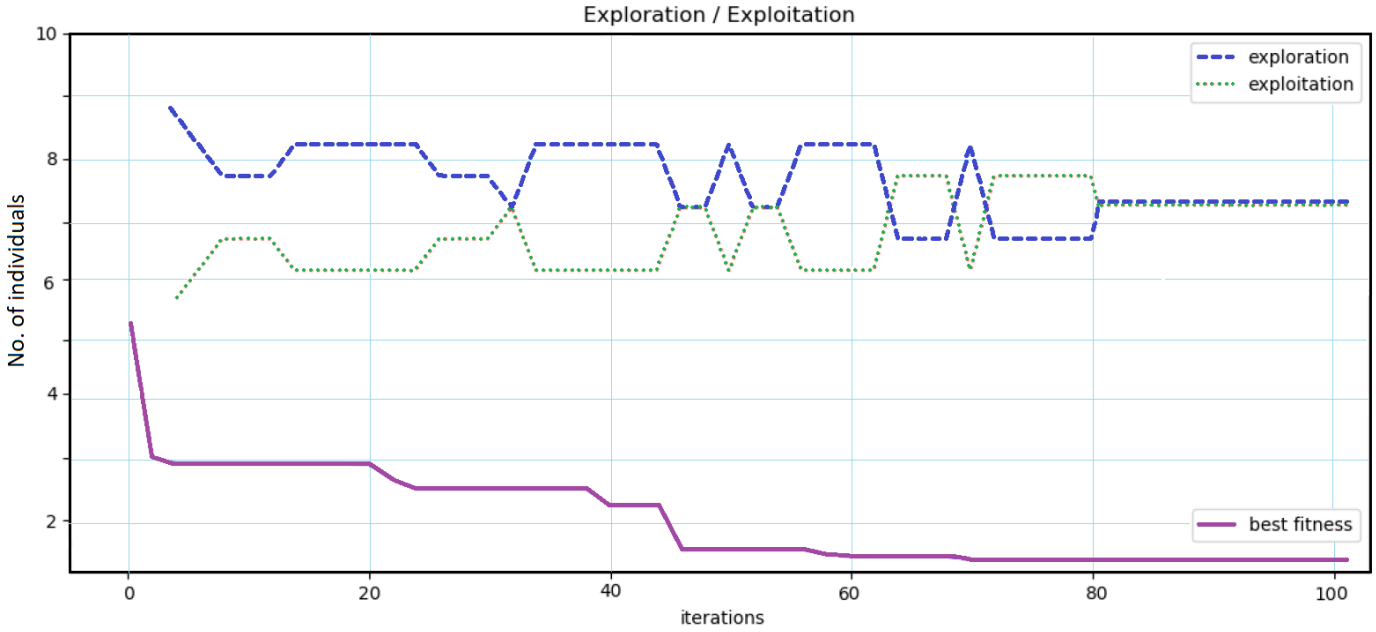


**Figure 4.** The exploration exploitation balance during the optimization process.

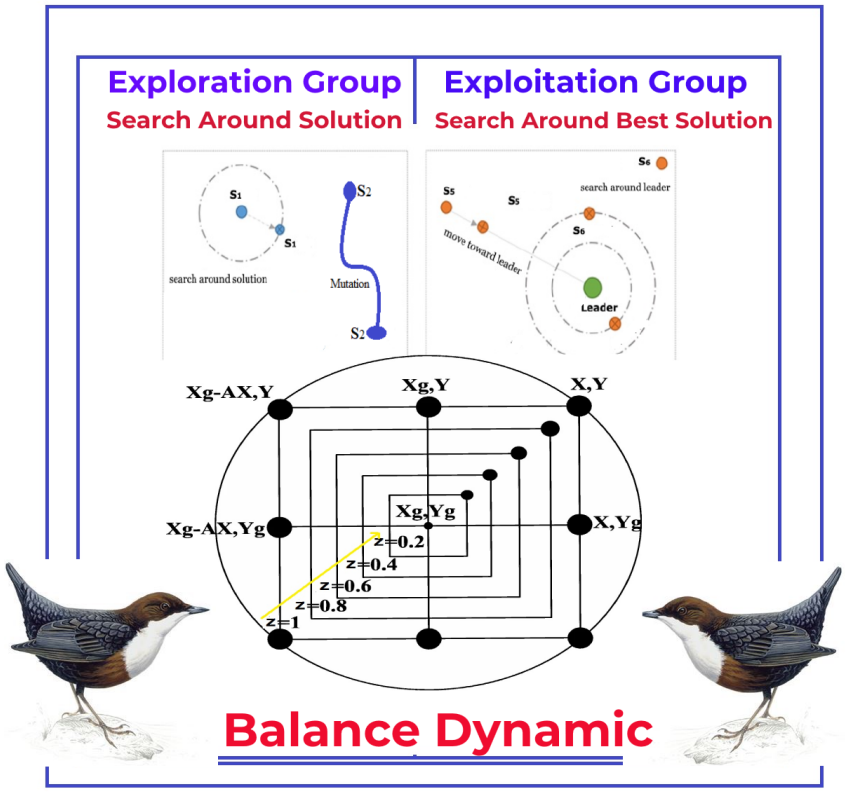


**Figure 5.** The balance dynamic between exploration and exploitation groups in the proposed optimization algorithm.

4.3.4. Binary Optimizer

In order to choose the best set of features to improve the classification accuracy, the output of the proposed DWDTO is transformed into binary {0, 1} using the sigmoid function represented by the following equation:

$$P_b^{(t+1)} = \begin{cases} 1 & \text{if } Sigmoid(P_{Best}) \geq 0.5 \\ 0 & otherwise \end{cases}, \quad Sigmoid(P_{Best}) = \frac{1}{1 + e^{-10(P_{Best} - 0.5)}} \tag{19}$$

where $P_{Best}$ refers to the best position, and t is the iteration number. The fitness function employed to measure the goodness of the candidate solution of selection features is represented by the following equation:

$$F_n = w_1 Error(P) + w_2 \frac{Number\ of\ selected\ features}{Total\ number\ of\ features} \tag{20}$$

where $P$ is a solution, $w_1 \in [0, 1]$, and $w_2 = 1 - w_1$, which are used to manage the importance of the number of the selected feature for population with size $n$ and the classification error rate. The steps of the proposed feature selection algorithm are presented in Algorithm 3.

**Algorithm 3** The proposed feature selection algorithm (binary bDWDTO)

**Input** population, size, fitness function
**Output** best set of features
**Initialize** Set DWDTO parameters
**Calculate** objective function and select best solutions
**Convert** solutions to binary {0, 1}
**Train** k-NN and calculate error
**Set** $t = 1, Max_{iter} = 100$
**while** $t \leq Max_{iter}$ **do**
    **Apply** DWDTO algorithm
    **Convert** solutions to binary
    **Calculate** Fitness
    **Update** Positions
**end while**
**Return** $X^*$

## 5. Experimental Results

The assessment of the proposed algorithms is performed in terms of two datasets. Firstly, the mobile KeyStroke Dynamics (KSD) Data Set [38] of which the first scenario denotes the experimental results. Secondly, the touch-screen-phone-based keystroke dynamics dataset [39], of which the second scenario represents the experimental results. The following sections discuss the achieved effects in both scenarios.

### *5.1. Evaluation Criteria*

The performance measures in Table 2 are used to assess how well the the proposed algorithms perform. For the first set of measurements, the performance of the feature selection process is measured. On the other hand, the second set of measures is utilized to evaluate the performance of the proposed optimized BRNN classification. The best solution at run $j$ is represented by $g_j^*$, and $size(g_j^*)$ refers to the size of the best solution vector. The number of optimizer runs is shown in the table as $M$. There are a total of $N$ points in the test set, and $C_i$ is the output label result from the used classifier. The point $i$'s

class label is $L_i$, and the total number of features is D. True Positive, True Negative, False Positive, and False Negative are referred to as $TP$, $TN$, $FP$, and $FN$, respectively.

**Table 2.** Metrics used in evaluating the proposed algorithms.

| Metric | Equation |
|---|---|
| Average fitness | $\frac{1}{M}\sum_{i=1}^{M} g_*^i$ |
| Worst Fitness | $max_{i=1}^{M} g_*^i$ |
| Best fitness | $min_{i=1}^{M} g_*^i$ |
| Average Error | $\frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} mse(C_i, L_i)$ |
| Average select size | $\frac{1}{M}\sum_{i=1}^{M} size(g_*^i)$ |
| Standard deviation | $\sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(g_*^i - Mean)^2}$ |
| Accuracy | $\frac{TP+TN}{TP+TN+FP+FN}$ |
| N-value (NPV) | $\frac{TN}{TN+FN}$ |
| *p*-value (PPV) | $\frac{TP}{TP+FP}$ |
| Sensitivity (TPR) | $\frac{TP}{TP+FN}$ |
| Specificity (TNR) | $\frac{TN}{TN+FP}$ |
| F1-Score | $\frac{TP}{TP+0.5(FP+FN)}$ |

### *5.2. Results of the First Scenario*

The first set of experiments recorded from the first dataset targets measuring the performance of the proposed feature selection algorithm. The results are presented in Table 3. In this table, the proposed bDWDTO algorithm achieves the best results compared to eleven other feature selection algorithms. These results show the superiority of the proposed feature selection algorithm.

**Table 3.** Evaluation results of the feature selection results achieved by the proposed algorithm and other competing algorithms when applied to the first dataset (D1).

| Algorithm | Avg. Error | Avg. Select Size | Avg. Fitness | Best Fitness | Worst Fitness | Std Fitness |
|---|---|---|---|---|---|---|
| bDWDTO | 0.510 | 0.654 | 0.537 | 0.442 | 0.637 | 0.345 |
| bGWO | 0.523 | 0.718 | 0.573 | 0.462 | 0.656 | 0.364 |
| bGWO_PSO | 0.516 | 0.713 | 0.556 | 0.539 | 0.617 | 0.348 |
| bPSO | 0.514 | 0.848 | 0.560 | 0.462 | 0.675 | 0.372 |
| bSFS | 0.522 | 0.672 | 0.574 | 0.523 | 0.597 | 0.364 |
| bWAO | 0.511 | 0.943 | 0.561 | 0.500 | 0.675 | 0.359 |
| bMGWO | 0.520 | 0.764 | 0.539 | 0.490 | 0.656 | 0.355 |
| bMVO | 0.511 | 0.818 | 0.561 | 0.520 | 0.636 | 0.352 |
| bSBO | 0.528 | 0.833 | 0.568 | 0.520 | 0.636 | 0.360 |
| bGWO_GA | 0.532 | 0.793 | 0.532 | 0.520 | 0.636 | 0.357 |
| bFA | 0.517 | 0.853 | 0.567 | 0.500 | 0.695 | 0.363 |
| bGA | 0.511 | 0.813 | 0.561 | 0.462 | 0.636 | 0.363 |

The convergence of the feature selection algorithm is recorded and represented by the plot shown in Figure 6. As shown in this figure, the proposed algorithm achieves the fastest convergence, which makes it superior to the other feature selection methods.

In addition, the time profile of the proposed feature selection algorithm compared to the other competing algorithms is presented in Table 4 for the datasets D1 and D2. In this table, the processing time using the proposed feature selection algorithm consumes the smallest timestamp, which gives another perspective of the superiority of the proposed algorithm.

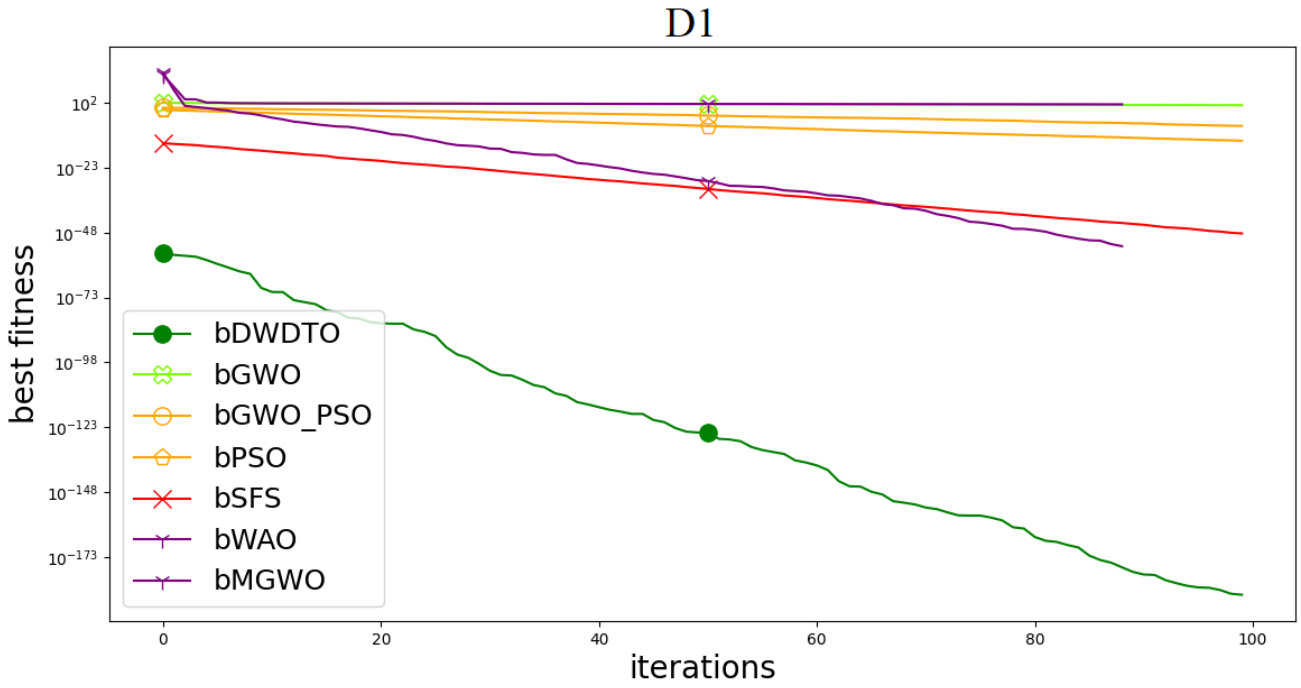


**Figure 6.** The convergence plot based on the proposed and other competing feature algorithms when applied to the first dataset (D1).

**Table 4.** Time profile (in seconds) of the proposed feature selection algorithm and other algorithms.

| Algorithm | D1 | D2 |
|---|---|---|
| bDWDTO | 12.534 | 12.952 |
| bGWO | 13.178 | 14.883 |
| bGWO_PSO | 12.77 | 14.02 |
| bPSO | 12.86 | 14.455 |
| bSFS | 14.26 | 14.21 |
| bWAO | 12.667 | 13.788 |
| bMGWO | 12.95 | 13.49 |
| bMVO | 13.121 | 14.395 |
| bSBO | 13.59 | 14.42 |
| bGWO_GA | 13.31 | 14.69 |
| bFA | 13.888 | 14.472 |
| bGA | 13.134 | 14.408 |

On the other hand, selecting the proper machine learning model is realized by evaluating the performance of three machine learning models and then determining the best performing model. Table 5 presents the results of user authentication using Neural Networks (NN), K-Nearest Neighbors (KNN), and BRNN. The best results are achieved using BRNN for all evaluation criteria and thus adopted for further steps of the proposed approach.

**Table 5.** Performance evaluation of the machine learning models applied to the first dataset (D1).

| Metric | NN | KNN | BRNN |
|---|---|---|---|
| Accuracy | 0.917 | 0.922 | 0.939 |
| Sensitivity (TPR) | 0.862 | 0.870 | 0.901 |
| Specificity (TNR) | 0.980 | 0.980 | 0.980 |
| Pvalue (PPV) | 0.980 | 0.980 | 0.980 |
| Nvalue (NPV) | 0.862 | 0.870 | 0.901 |
| F-score | 0.917 | 0.922 | 0.939 |
| Time (seconds) | 137 | 125 | 102 |

On the other hand, the evaluation of the proposed DWDTO algorithm is measured in terms of optimizing the BRNN network and analyzing the results. Table 6 presents the statistical analysis of the recorded results based on the proposed optimization algorithm and five other optimization algorithms. In this table, the proposed optimization algorithms are proven to achieve better performance than the other competing algorithms for the 20 samples in the test set. The maximum accuracy achieved by the proposed algorithm is (98.89%), whereas the best accuracy achieved by the other algorithms is (96.12%), which is achieved by GWO when used to optimize the BRNN model.

**Table 6.** Statistical analysis of the achieved results using the proposed DWDTO applied to the first dataset (D1).

| Metric | DWDTO + BRNN | GWO | WOA | PSO | GA | GSA |
|---|---|---|---|---|---|---|
| Num. Values | 20 | 20 | 20 | 20 | 20 | 20 |
| Minimum | 0.9889 | 0.9612 | 0.9378 | 0.9598 | 0.9523 | 0.9563 |
| 25% | 0.9900 | 0.9712 | 0.9578 | 0.9685 | 0.9623 | 0.9563 |
| Median | 0.9900 | 0.9712 | 0.9578 | 0.9685 | 0.9623 | 0.9563 |
| 75% | 0.9900 | 0.9712 | 0.9578 | 0.9685 | 0.9623 | 0.9563 |
| Maximum | 0.9927 | 0.9812 | 0.9698 | 0.9798 | 0.9723 | 0.9763 |
| Range | 0.0038 | 0.0200 | 0.0320 | 0.0200 | 0.0200 | 0.0200 |
| Mean | 0.9901 | 0.9712 | 0.9574 | 0.9686 | 0.9623 | 0.9582 |
| Std. | 0.0007 | 0.0032 | 0.0053 | 0.0033 | 0.0032 | 0.0050 |
| Std. Error | 0.0001 | 0.0007 | 0.0012 | 0.0007 | 0.0007 | 0.0011 |
| Skewness | 3.289 | $5.703 \times 10^{-14}$ | −2.171 | 1.263 | 0 | 3.014 |
| Kurtosis | 14.79 | 9.5 | 11.74 | 10.25 | 9.5 | 9.335 |
| Sum | 19.8 | 19.42 | 19.15 | 19.37 | 19.25 | 19.16 |

The significance of the proposed algorithm is measured using the Wilcoxon signed rank test and one-way Analysis of Variance (ANOVA) test. The results of these tests are presented in Tables 7 and 8, respectively. To prove that the proposed DWDTO algorithm is significantly different from all the other competing algorithms, we calculate the $p$-values between the two algorithms. This study uses Wilcoxon's rank-sum test. This test has two basic hypotheses: the null and the alternative hypotheses. $\mu DWDTO = \mu GWO, \mu DWDTO = \mu WOA, \mu DWDTO = \mu PSO, \mu DWDTO = \mu GA$, and $\mu DWDTO = \mu GSA$ are the mean, $\mu$, values of the null hypothesis represented by H0. However, the H1 hypothesis does not take into account the algorithms' capabilities when comparing the results. The Wilcoxon rank-sum test results are shown in Table 7. The suggested algorithm's $p$-values are less than 0.05 compared to those of the other methods. These findings show that the suggested feature selection strategy is better and statistically significant. The ANOVA test examines the statistical differences between the proposed DWDTO algorithm and the other methods. The null hypothesis's mean values, indicated by H0, include $\mu DWDTO = \mu GWO = \mu WOA = \mu PSO = \mu GA = \mu GSA$. Table 8 shows the results of the ANOVA test as assessed. As illustrated in Figure 7, we have the residual plot, followed by the homoscedasticity plot and the QQ plots. The residual error and homoscedasticity values are both within -0.03 and +0.025, as shown in the plots of the figure, indicating the robustness of the suggested technique. The QQ figure also demonstrates that projected outcomes match actual values, which confirms that the proposed algorithm is robust. The heat map plot, on the other hand, indicates how the algorithm stacks up against other competing methods in the literature.

**Table 7.** Wilcoxon signed-rank test of the achieved results using the first dataset (D1).

| Metric | DWDTO + BRNN | GWO | WOA | PSO | GA | GSA |
|---|---|---|---|---|---|---|
| Theo. median | 0 | 0 | 0 | 0 | 0 | 0 |
| Act. median | 0.99 | 0.9712 | 0.9578 | 0.9685 | 0.9623 | 0.9563 |
| Num. Values | 20 | 20 | 20 | 20 | 20 | 20 |
| Sum ranks | 210 | 210 | 210 | 210 | 210 | 210 |
| Sum +ranks | 210 | 210 | 210 | 210 | 210 | 210 |
| Sum −ranks | 0 | 0 | 0 | 0 | 0 | 0 |
| *p* value | <0.0001 | <0.0001 | <0.0001 | <0.0001 | <0.0001 | <0.0001 |
| Significance | Yes | Yes | Yes | Yes | Yes | Yes |
| Discrepancy | 0.99 | 0.9712 | 0.9578 | 0.9685 | 0.9623 | 0.9563 |

**Table 8.** One-way analysis of variance test of the achieved results using the first dataset (D1).

| | SS | DF | MS | F (DFn, DFd) | *p* Value |
|---|---|---|---|---|---|
| Treatment | 0.01246 | 4 | 0.003115 | F (4, 95) = 256.8 | $p < 0.0001$ |
| Residual | 0.001152 | 95 | 0.00001213 | | |
| Total | 0.01361 | 99 | | | |

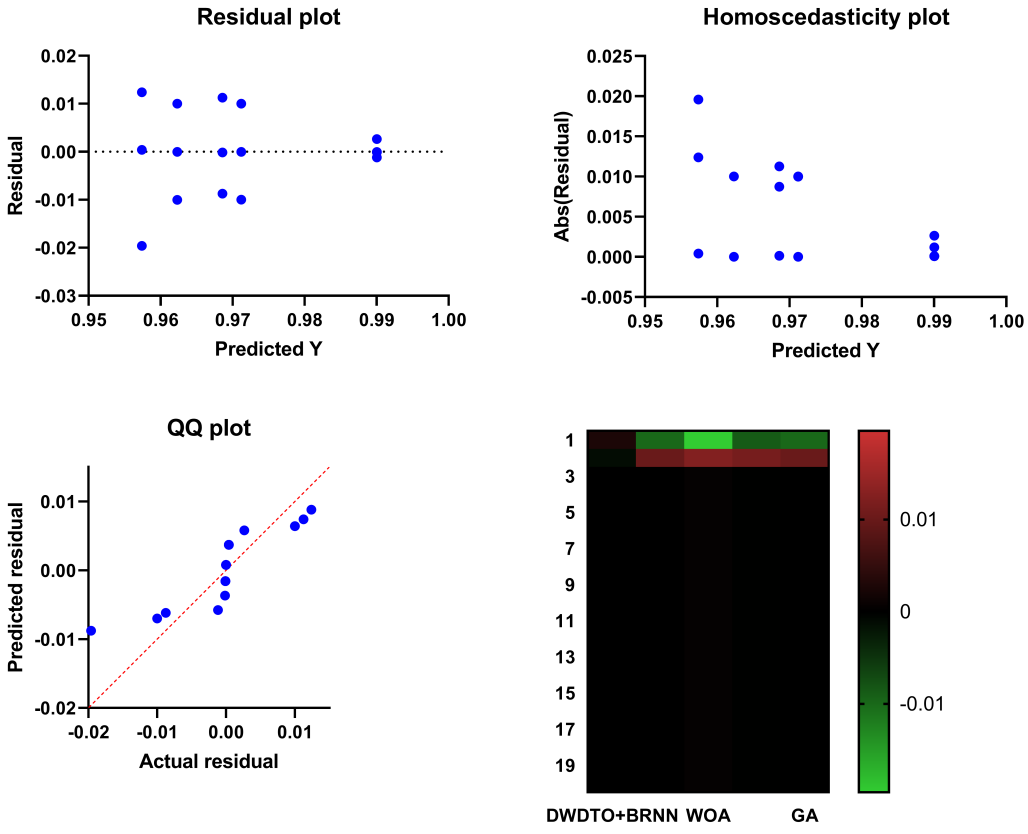


**Figure 7.** Visual representation of the results achieved by the proposed optimization algorithm in terms of the first dataset (D1).

### *5.3. Results of the Second Scenario*

Similarly, the evaluation of the proposed algorithms using the second dataset, D2, is performed in terms of the feature selection and the optimization algorithms. The results of the assessment of the feature selection algorithm are presented in Table 9 based on the adopted evaluation criteria. In addition, the convergence curve of the feature selection using the proposed and other competing algorithms is shown in Figure 8. These results

emphasize the superiority of the proposed feature selection algorithms in terms of the second dataset.

**Table 9.** Evaluation results of the feature selection results achieved by the proposed and other competing algorithms when applied to the second dataset (D2).

| Algorithm | Avg. Error | Avg. Select Size | Avg. Fitness | Best Fitness | Worst Fitness | Std Fitness |
|---|---|---|---|---|---|---|
| bDWDTO | 0.447 | 0.449 | 0.459 | 0.407 | 0.558 | 0.345 |
| bGWO | 0.460 | 0.579 | 0.494 | 0.424 | 0.568 | 0.355 |
| bGWO_PSO | 0.461 | 0.603 | 0.461 | 0.449 | 0.517 | 0.351 |
| bPSO | 0.488 | 0.803 | 0.521 | 0.458 | 0.576 | 0.346 |
| bSFS | 0.467 | 0.627 | 0.467 | 0.414 | 0.600 | 0.387 |
| bWAO | 0.473 | 0.644 | 0.507 | 0.433 | 0.593 | 0.355 |
| bMGWO | 0.449 | 0.567 | 0.491 | 0.457 | 0.576 | 0.350 |
| bMVO | 0.482 | 0.784 | 0.515 | 0.416 | 0.559 | 0.352 |
| bSBO | 0.468 | 0.743 | 0.468 | 0.441 | 0.543 | 0.350 |
| bGWO_GA | 0.507 | 0.737 | 0.507 | 0.492 | 0.602 | 0.359 |
| bFA | 0.478 | 0.800 | 0.512 | 0.407 | 0.610 | 0.360 |
| bGA | 0.468 | 0.703 | 0.502 | 0.441 | 0.619 | 0.360 |

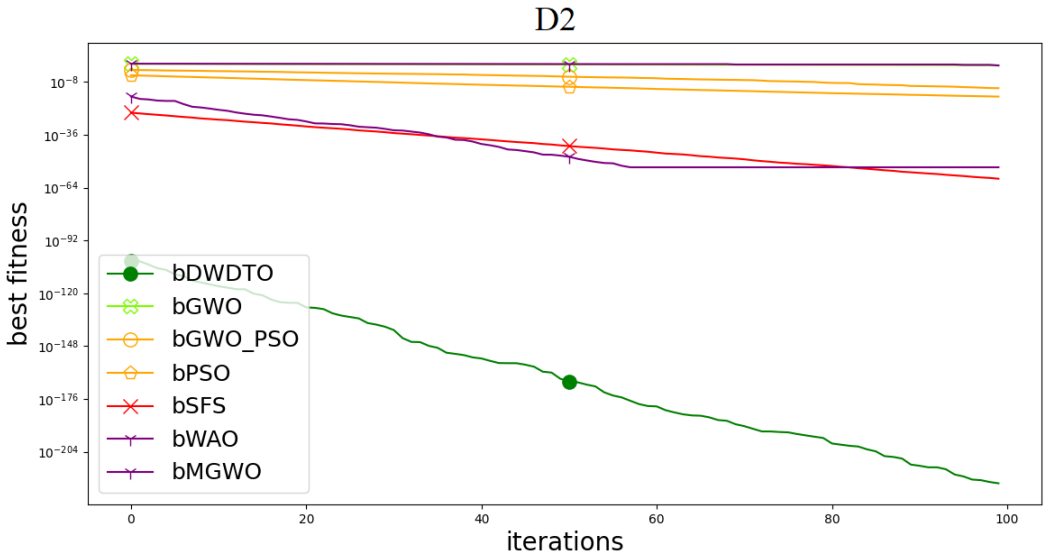


**Figure 8.** The convergence plot based on the proposed and other competing feature algorithms when applied to the second dataset (D2).

Moreover, the best machine learning model that fits the keystroke data in the second dataset is applied by evaluating three machine learning models and selecting the best-performing model. Table 10 presents the evaluation of the three models. In this table, the best performing model is BRNN and thus selected for optimization and categorizing smartphone users based on keystroke data.

**Table 10.** Evaluation of the performance of machine learning models applied to the second dataset (D2).

| Metric | NN | KNN | BRNN |
|---|---|---|---|
| Accuracy | 0.932 | 0.941 | 0.955 |
| Sensitivity (TPR) | 0.857 | 0.895 | 0.895 |
| Specificity (TNR) | 0.989 | 0.989 | 0.993 |
| *p*-value (PPV) | 0.984 | 0.988 | 0.988 |
| N-value (NPV) | 0.900 | 0.900 | 0.938 |
| F-score | 0.916 | 0.939 | 0.939 |
| Time (seconds) | 118 | 107 | 97 |

Similar to the analysis performed on the results of optimizing BRNN using the proposed algorithm on the first dataset, the same analysis is performed on the results achieved on the second dataset (D2). Tables 11–13 present the statistical analysis, Wilcoxon signed-rank test and ANOVA test. The results presented in these tables confirm the superiority and effectiveness of the proposed optimization algorithm.

**Table 11.** Statistical analysis of the achieved results using the proposed DWDTO applied to the second dataset (D2).

| Metric | DWDTO + BRNN | GWO | WOA | PSO | GA | GSA |
|---|---|---|---|---|---|---|
| Num. Values | 20 | 20 | 20 | 20 | 20 | 20 |
| Minimum | 0.9899 | 0.9689 | 0.9465 | 0.9599 | 0.9700 | 0.9471 |
| 25% | 0.9934 | 0.9789 | 0.9665 | 0.9689 | 0.9800 | 0.9571 |
| Median | 0.9934 | 0.9789 | 0.9665 | 0.9689 | 0.9800 | 0.9571 |
| 75% | 0.9934 | 0.9789 | 0.9680 | 0.9689 | 0.9800 | 0.9571 |
| Maximum | 0.9934 | 0.9889 | 0.9767 | 0.9729 | 0.9900 | 0.9771 |
| Range | 0.0034 | 0.0200 | 0.0301 | 0.0130 | 0.0200 | 0.0300 |
| Mean | 0.9932 | 0.9788 | 0.9673 | 0.9686 | 0.9800 | 0.9581 |
| Std. | 0.0008 | 0.0033 | 0.0061 | 0.0023 | 0.0032 | 0.0055 |
| Std. Error | 0.0002 | 0.0007 | 0.0014 | 0.0005 | 0.0007 | 0.0012 |
| Skewness | −4.472 | 0.0496 | −1.694 | −2.964 | $-5.703 \times 10^{-14}$ | 2.164 |
| Kurtosis | 20 | 9.379 | 7.402 | 13.36 | 9.5 | 8.21 |
| Sum | 19.86 | 19.58 | 19.35 | 19.37 | 19.6 | 19.16 |

**Table 12.** Wilcoxon signed-rank test of the achieved results using the second dataset (D2).

| Metric | DWDTO + BRNN | GWO | WOA | PSO | GA | GSA |
|---|---|---|---|---|---|---|
| Theo. median | 0 | 0 | 0 | 0 | 0 | 0 |
| Act. median | 0.9934 | 0.9789 | 0.9665 | 0.9689 | 0.98 | 0.96 |
| Num. Values | 20 | 20 | 20 | 20 | 20 | 20 |
| Sum ranks | 210 | 210 | 210 | 210 | 210 | 210 |
| Sum +ranks | 210 | 210 | 210 | 210 | 210 | 210 |
| Sum −ranks | 0 | 0 | 0 | 0 | 0 | 0 |
| *p*-value | <0.0001 | <0.0001 | <0.0001 | <0.0001 | <0.0001 | <0.0001 |
| Significance | Yes | Yes | Yes | Yes | Yes | Yes |
| Discrepancy | 0.9934 | 0.9789 | 0.9665 | 0.9689 | 0.98 | 0.96 |

**Table 13.** One-way analysis of variance test of the achieved results using the second dataset (D2).

| | SS | DF | MS | F (DFn, DFd) | *p* Value |
|---|---|---|---|---|---|
| Treatment | 0.008733 | 4 | 0.002183 | F (4, 95) = 170.9 | $p < 0.0001$ |
| Residual | 0.001214 | 95 | 0.00001278 | | |
| Total | 0.009946 | 99 | | | |

Moreover, the visual plots shown in Figure 9 represent the performance of the proposed algorithm in categorizing smartphone users based on the second dataset (D2). These plots are the residual, homoscedasticity, QQ, and heatmap. The results represented by these plots emphasize the effectiveness of the proposed optimization algorithm.

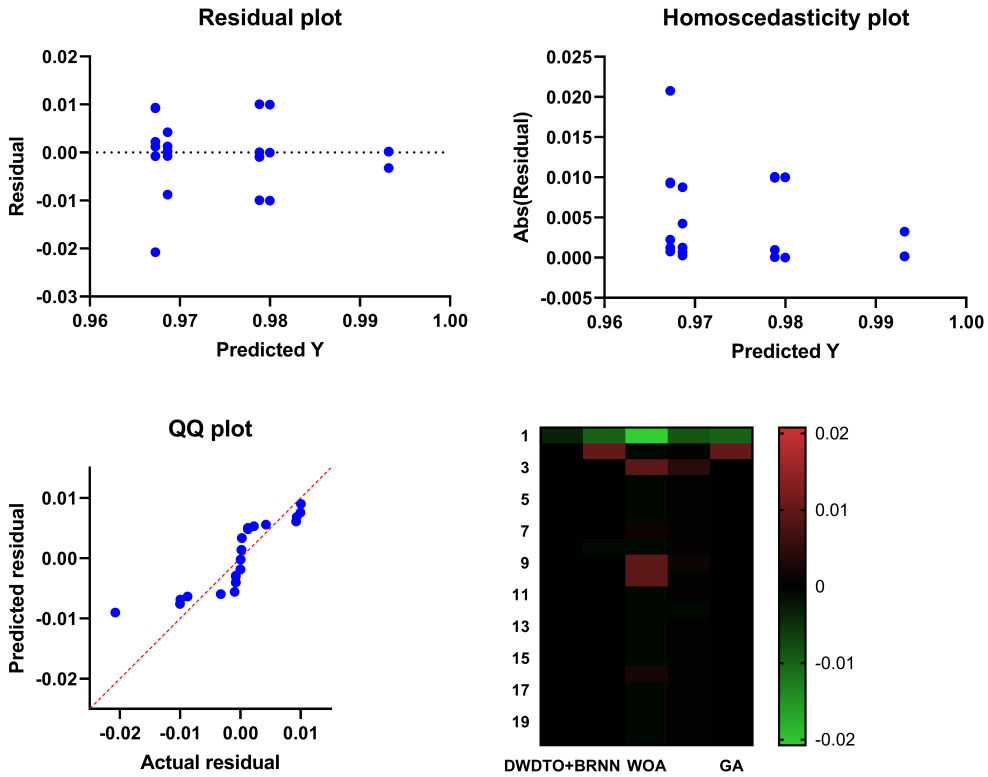


**Figure 9.** Visual representation of the results achieved by the proposed optimization algorithm in terms of the second dataset (D2).

### *5.4. Classification Results*

The classification results using the optimized BRNN are recorded and measured in terms of the adopted evaluation criteria mentioned in the previous section. The results are presented in Table 14. As shown in this table, the proposed optimization algorithm could achieve a promising performance with accuracy greater than (99%) on both D1 and D2. The other evaluation criteria recorded in the table confirm the effectiveness of the proposed algorithm.

**Table 14.** The performance of the BRNN after optimization using the proposed DWDTO applied to the two datasets, D1 and D2.

| Metric | D1 | D2 |
|---|---|---|
| Accuracy | 0.990182803 | 0.993208829 |
| Sensitivity (TRP) | 0.946547884 | 0.965909091 |
| Specificity (TNP) | 0.998003992 | 0.998003992 |
| *p*-value (PPV) | 0.988372093 | 0.988372093 |
| N-value (NPV) | 0.990491284 | 0.994035785 |
| F-Score | 0.967007964 | 0.977011494 |
| Time (seconds) | 77 | 59 |

### *5.5. Comparison with Other Methods*

To emphasize the superiority of the proposed optimization algorithm, a set of experiments was conducted to prove the expected findings. Figure 10 shows the average time over the two datasets. In this figure, the proposed feature selection algorithm achieves the shortest average time, represented by a red star.

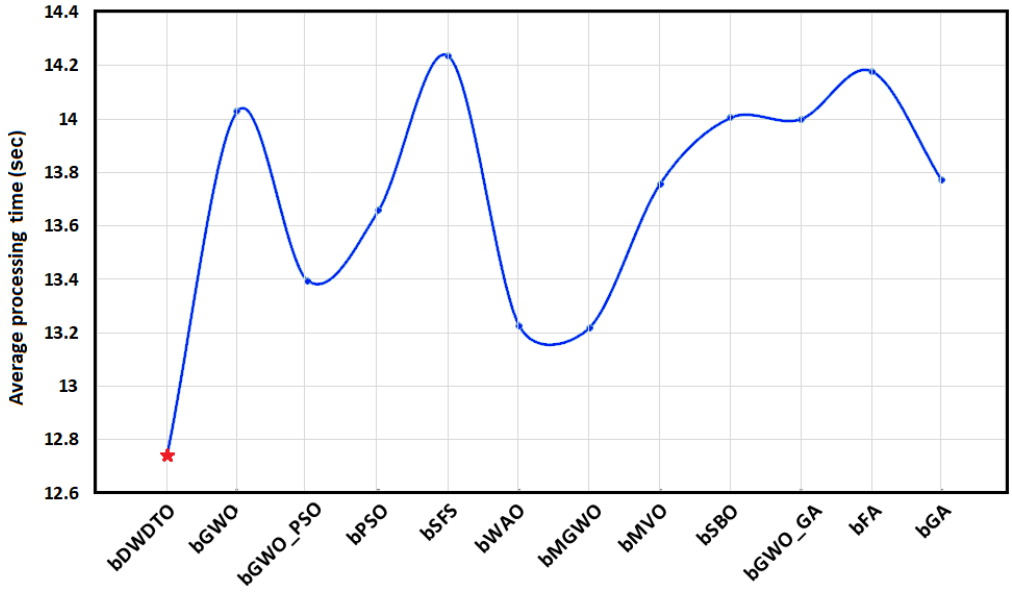


**Figure 10.** Average processing time (in seconds) using the proposed optimization algorithm and the other competing optimization algorithms. The red star refers to the minimum average processing time which is achieved by the proposed bDWDTO algorithm.

In addition, Figures 11 and 12 show the histograms of the accuracy achieved by optimizing BRNN using the proposed optimization algorithm and other five optimization algorithms on both D1 and D2. As shown in this figure, the proposed algorithm achieves the best performance for almost all the test samples. These results show the superiority of the proposed optimization algorithm.

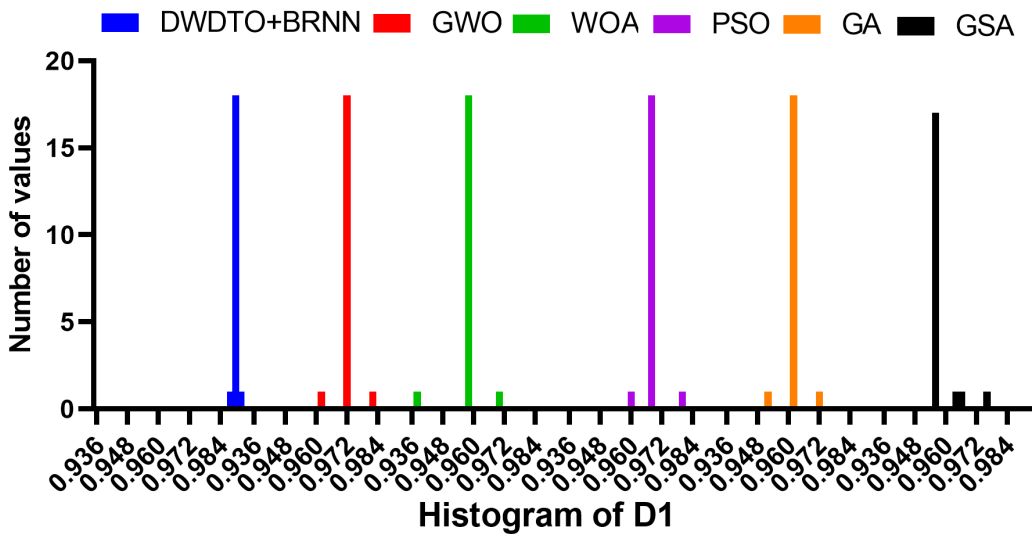


**Figure 11.** Histogram of the accuracy achieved by the proposed approach and other optimization approaches in terms of the first dataset (D1).

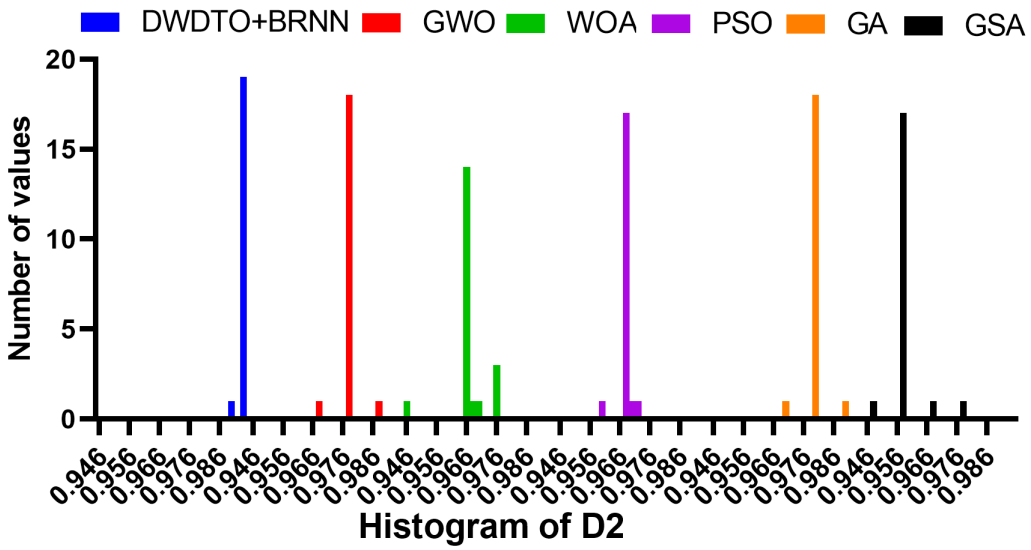


**Figure 12.** Histogram of the accuracy achieved by the proposed approach and other optimization approaches in terms of the second dataset (D2).

Moreover, the plots shown in Figure 13 present an additional comparison between the proposed optimization algorithm and other algorithms in terms of the accuracy and ROC curves. These plots confirm the expected findings and prove the superiority of the proposed algorithm.

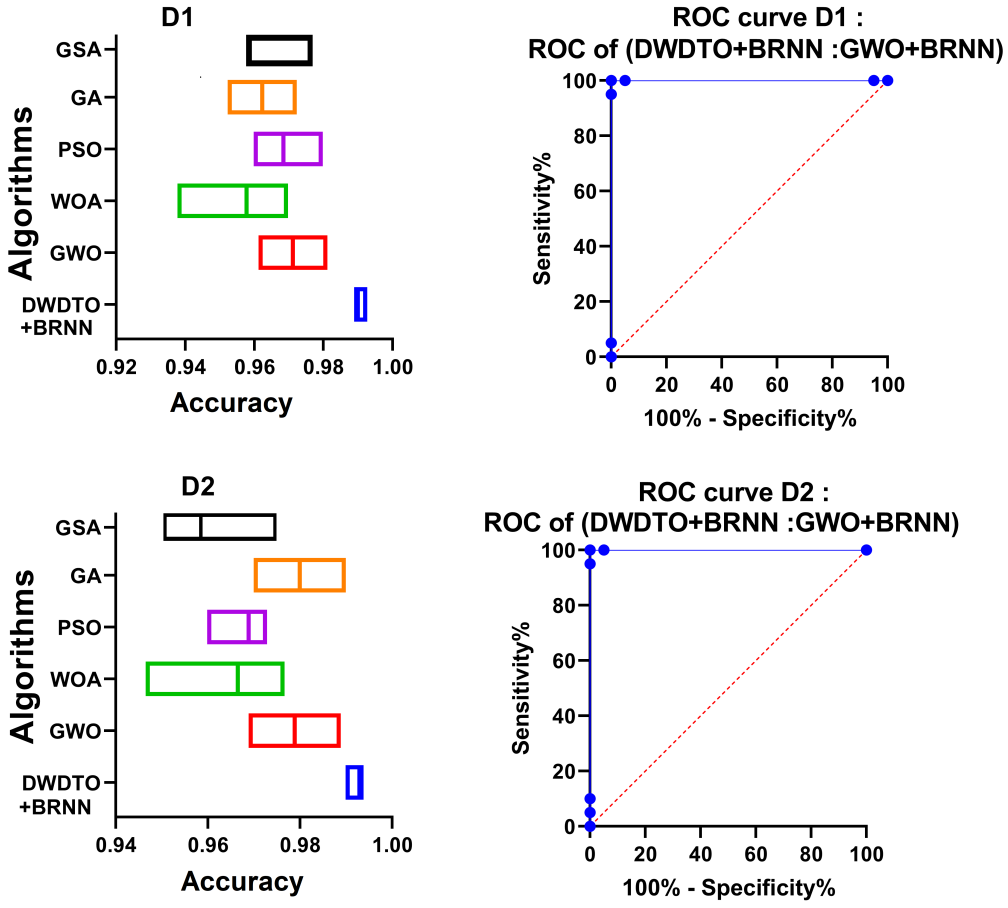


**Figure 13.** Comparison between the performance of the proposed optimization algorithm and the other competing algorithms applied to the two datasets, D1 and D2.

## 6. Conclusions

Keystroke-dynamics-based smartphone user authentication can by improved by using meta-heuristic optimization. Using the proposed optimization algorithm, referred to as DWDTO, a Bidirectional Recurrent Neural Network is optimized to boost the categorization accuracy of smartphone users. The optimization algorithm was put to the test to see if it could accurately reflect the authenticated user while also performing better than standard optimization methods in terms of user categorization. In order to evaluate the categorization performance of the proposed algorithm, two datasets were compared. Three classifiers were also assessed for utilizing the optimization algorithm to evaluate their user categorization performance. The data features obtained via smartphone and feature selection algorithm were all set to the same for accurate comparison of categorization performance with other competing methods. Consequently, it was discovered that the proposed optimization algorithm provided superior classification performance over the existing methods. Classification of datasets using the optimized classifier was validated. The data collection approach described in this study has a high potential for use as one of the data collecting methods for keystroke-data-based authentication systems based on the principal findings of this investigation. In addition, a statistical analysis is performed to measure the stability and effectiveness of the proposed algorithms. The results confirm the superiority and stability of the proposed algorithms when compared with other competing algorithms.

**Author Contributions:** Formal analysis, E.-S.M.E.-K. and A.A.A.; Investigation, N.K.; Methodology, A.I.; Resources, S.M.; Visualization, M.M.E. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Abualigah, L.; Elaziz, M.A.; Khodadadi, N.; Forestiero, A.; Jia, H.; Gandomi, A.H. Aquila Optimizer Based PSO Swarm Intelligence for IoT Task Scheduling Application in Cloud Computing. In *Integrating Meta-Heuristics and Machine Learning for Real-World Optimization Problems*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 481–497.
2. Sharma, R.; Sharma, V.K.; Singh, A. A Review Paper on Facial Recognition Techniques. In Proceedings of the 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 11–13 November 2021; pp. 617–621.
3. Ali, M.M.; Mahale, V.H.; Yannawar, P.; Gaikwad, A.T. Overview of fingerprint recognition system. In Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 1334–1338.
4. Daniel, D.M.; Monica, B. Person authentication technique using human iris recognition. In Proceedings of the 2010 9th International Symposium on Electronics and Telecommunications, Timisoara, Romania, 11–12 November 2010; pp. 265–268.
5. Kaveh, A.; Eslamlou, A.D.; Khodadadi, N. Dynamic water strider algorithm for optimal design of skeletal structures. *Period. Polytech. Civ. Eng.* **2020**, *64*, 904–916. [CrossRef]
6. Kaveh, A.; Talatahari, S.; Khodadadi, N. Stochastic paint optimizer: Theory and application in civil engineering. *Eng. Comput.* **2022**, *38*, 1921–1952. [CrossRef]
7. Kaveh, A.; Khodadadi, N.; Talatahari, S. A comparative study for the optimal design of steel structures using CSS and ACSS algorithms. *Int. J. Optim. Civ. Eng* **2021**, *11*, 31–54.
8. Khodadadi, N.; Abualigah, L.; Mirjalili, S. Multi-objective Stochastic Paint Optimizer (MOSPO). *Neural Comput. Appl.* **2022**, *2022*, 1–24. doi: 10.1007/s00521-022-07405-z. [CrossRef]
9. Khodadadi, N.; Mirjalili, S. Truss optimization with natural frequency constraints using generalized normal distribution optimization. *Appl. Intell.* **2022**, *52*, 10384–10397. [CrossRef]
10. Kaveh, A.; Talatahari, S.; Khodadadi, N. The Hybrid Invasive Weed Optimization-Shuffled Frog-leaping Algorithm Applied to Optimal Design of Frame Structures. *Period. Polytech. Civ. Eng.* **2019**, *63*, 882–897. [CrossRef]

11. Ryu, Y.S.; Koh, D.H.; Aday, B.L.; Gutierrez, X.A.; Platt, J.D. Usability Evaluation of Randomized Keypad. *J. Usability Study* **2010**, *5*, 65–75.
12. Spillane, R. Keyboard apparatus for personal identification. *IBM Tech. Discl. Bull.* **1975**, *17*, 3346.
13. Umphress, D.; Williams, G. Identity verification through keyboard characteristics. *Int. J. Man-Mach. Stud.* **1985**, *23*, 263–273. [CrossRef]
14. Campisi, P.; Maiorana, E.; Lo Bosco, M.; Neri, A. User authentication using keystroke dynamics for cellular phones. *IET Signal Process.* **2009**, *3*, 333. [CrossRef]
15. Lee, H.; Hwang, J.Y.; Kim, D.I.; Lee, S.; Lee, S.H.; Shin, J.S. Understanding Keystroke Dynamics for Smartphone Users Authentication and Keystroke Dynamics on Smartphones Built-In Motion Sensors. *Secur. Commun. Netw.* **2018**, *2018*, 2567463. [CrossRef]
16. Cockell, R.; Halak, B. On the Design and Analysis of a Biometric Authentication System Using Keystroke Dynamics. *Cryptography* **2020**, *4*, 12. [CrossRef]
17. Alsultan, A.; Warwick, K.; Wei, H. Non-conventional keystroke dynamics for user authentication. *Pattern Recognit. Lett.* **2017**, *89*, 53–59. [CrossRef]
18. Kim, J.; Kang, P. Freely typed keystroke dynamics-based user authentication for mobile devices based on heterogeneous features. *Pattern Recognit.* **2020**, *108*, 107556. [CrossRef]
19. Kiyani, A.T.; Lasebae, A.; Ali, K.; Rehman, M.U.; Haq, B. Continuous User Authentication Featuring Keystroke Dynamics Based on Robust Recurrent Confidence Model and Ensemble Learning Approach. *IEEE Access* **2020**, *8*, 156177–156189. [CrossRef]
20. Porwik, P.; Doroz, R.; Wesolowski, T.E. Dynamic keystroke pattern analysis and classifiers with competence for user recognition. *Appl. Soft Comput.* **2021**, *99*, 106902. [CrossRef]
21. Saini, B.S.; Singh, P.; Nayyar, A.; Kaur, N.; Bhatia, K.S.; El-Sappagh, S.; Hu, J.W. A Three-Step Authentication Model for Mobile Phone User Using Keystroke Dynamics. *IEEE Access* **2020**, *8*, 125909–125922. [CrossRef]
22. Jalaly Bidgoly, A.; Jalaly Bidgoly, H.; Arezoumand, Z. A survey on methods and challenges in EEG based authentication. *Comput. Secur.* **2020**, *93*, 101788. [CrossRef]
23. Ingale, M.; Cordeiro, R.; Thentu, S.; Park, Y.; Karimian, N. ECG Biometric Authentication: A Comparative Analysis. *IEEE Access* **2020**, *8*, 117853–117866. [CrossRef]
24. Maiti, A.; Crager, K.; Jadliwala, M.; He, J.; Kwiat, K.; Kamhoua, C. RandomPad: Usability of Randomized Mobile Keypads for Defeating Inference Attacks. In Proceedings of the IEEE Euro Workshop on Innovations in Mobile Privacy & Security (IMPS), Paris, France, 29 April 2017; pp. 1–10.
25. Benjapatanamongkol, N.; Bhattarakosol, P. A Preliminary Study of Finger Area and Keystroke Dynamics Using Numeric Keypad With Random Numbers on Android Phones. In Proceedings of the 2019 23rd International Computer Science and Engineering Conference (ICSEC), Phuket, Thailand, 30 October–1 November 2019; pp. 30–34.
26. Yu, E.; Cho, S. GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification. In Proceedings of the International Joint Conference on Neural Networks, 2003, Portland, OR, USA, 20–24 July 2003; Volume 3, pp. 2253–2257.
27. Azevedo, G.L.F.B.G.; Cavalcanti, G.D.C.; Carvalho Filho, E.C.B. An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 3577–3584.
28. Karnan, M.; Muthuramalingam, A.; Kalamani, A. Feature subset selection in keystroke dynamics using ant colony optimization. *J. Eng. Technol. Res.* **2009**, *1*, 72–80.
29. Karnan, M.; Akila, M. Personal Authentication Based on Keystroke Dynamics Using Soft Computing Techniques. In Proceedings of the 2010 Second International Conference on Communication Software and Networks, Singapore, 26–28 February 2010; pp. 334–338.
30. Solami, E.A.; Boyd, C.; Clark, A.; Ahmed, I. User-representative feature selection for keystroke dynamics. In Proceedings of the 2011 5th International Conference on Network and System Security, Milan, Italy, 6–8 September 2011; pp. 229–233.
31. El-Kenawy, E.S.M.; Mirjalili, S.; Alassery, F.; Zhang, Y.D.; Eid, M.M.; El-Mashad, S.Y.; Aloyaydi, B.A.; Ibrahim, A.; Abdelhamid, A.A. Novel Meta-Heuristic Algorithm for Feature Selection, Unconstrained Functions and Engineering Problems. *IEEE Access* **2022**, *10*, 40536–40555. [CrossRef]
32. Abdelhamid, A.A.; El-Kenawy, E.S.M.; Alotaibi, B.; Amer, G.M.; Abdelkader, M.Y.; Ibrahim, A.; Eid, M.M. Robust Speech Emotion Recognition Using CNN+LSTM Based on Stochastic Fractal Search Optimization Algorithm. *IEEE Access* **2022**, *10*, 49265–49284. [CrossRef]
33. Sami Khafaga, D.; Ali Alhussan, A.; El-kenawy, E.S.M.; Takieldeen, A.E.; Hassan, T.M.; Hegazy, E.A.; Eid, E.A.F.; Ibrahim, A.; Abdelhamid, A.A. Meta-heuristics for Feature Selection and Classification in Diagnostic Breast-Cancer. *Comput. Mater. Contin.* **2022**, *73*, 749–765.
34. Choi, M.; Lee, S.; Jo, M.; Shin, J.S. Keystroke dynamics-based authentication using unique keypad. *Sensors* **2021**, *21*, 2242. [CrossRef] [PubMed]
35. Sami Khafaga, D.; Ali Alhussan, A.; El-kenawy, E.S.M.; Ibrahim, A.; Abd Elkhalik, S.H.; El-Mashad, S.Y.; Abdelhamid, A.A. Improved Prediction of Metamaterial Antenna Bandwidth Using Adaptive Optimization of LSTM. *Comput. Mater. Contin.* **2022**, *73*, 865–881. [CrossRef]

36. Abdel Samee, N.; El-Kenawy, E.S.M.; Atteia, G.; Jamjoom, M.M.; Ibrahim, A.; Abdelhamid, A.A.; El-Attar, N.E.; Gaber, T.; Slowik, A.; Shams, M.Y. Metaheuristic Optimization Through Deep Learning Classification of COVID-19 in Chest X-Ray Images. *Comput. Mater. Contin.* **2022**, *73*, 4193–4210.
37. Nasser AlEisa, H.; El-kenawy, E.S.M.; Ali Alhussan, A.; Saber, M.; Abdelhamid, A.A.; Sami Khafaga, D. Transfer Learning for Chest X-rays Diagnosis Using Dipper Throated Algorithm. *Comput. Mater. Contin.* **2022**, *73*, 2371–2387.

38. MEU-Mobile KSD Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/MEU-Mobile+KSD (accessed on 28 June 2022).
39. RHU KeyStroke Dynamics Benchmark Dataset. Available online: https://www.coolestech.com/rhu-keystroke/ (accessed on 28 June 2022).